Java HLA RTI API

Richard Weatherly, Ph.D.

http://ms.ie.org/weatherly

February 13, 1997

Goals: Intended to Look Familiar

- **Structurally similar to C++ and Ada APIs**
- **Idiomatic to Java**
- **Neutral with respect to process model**
- **Follows I/F Specification 1.0**

The Ambassadors Are Defined As In C++

- **The RTIambassador**
  - **All federate initiated services are methods on the RTIambassador.**
  - **RTIambassador is instantiated before any services can be invoked.**
  - **One RTIambassador instance for each federate instance in a given federation execution.**
- **The Federate**
  - **The federate is represented by a single interface, Federate, which the federate developer must implement.**
  - **Reference to the Federate implementation is a parameter on the joinFederationExecution service.**

Avoid Java Types Where They Add No Value

- **All Java types are classes, unlike C++ or Ada.**
- **Where user will manipulate a type as a primitive, define it as the Java primitive type. For example:**
  - **Names are Strings.**
  - **Time is a double.**
  - **The assumption is that this lack of type restriction won't lead to errors in practice.**
- **ObjectIDs have been defined as an integer.**
  - **Subject of some discussion**
  - **Conceptually an opaque type**
  - **In practice used as hash or array indices**
  - **I/F Specification suggests they exist in ranges**

Create Java Types Where Useful

- **Exceptions are Java Exceptions.**
  - **All the C++ exceptions are there.**
  - **All derive from a base class RTIexception.**
  - **They carry a serial number and "reason" String inherited from the Java Exception class.**
- **Structures must be classes:**
  - **To return a record, use a class with public members, e.g., ObjectIDrange.**
  - **Handle-value pairs**
  - **Sets of handles, sets of HVPs**
- **Opaque types must be classes:**
  - **"Enumerated types"**
  - **Handles (attribute, parameter, class)**

Handles Are Opaque Types

- **Each handle class is an interface to the public, and an implementation class known to the RTI.**
- **RTI creates values of handles and passes them to federates.**
- **Federate operations on handles are limited:**
  - **Has no access to a handle's intrinsic value**
  - **Can use handles as keys for hash tables**
  - **Can compare handles of same type for equality**
- **RTI has access to intrinsic value and uses it in set classes.**
  - **Because the value of many handles is limited, set classes use handle values as array indices.**
  - **We may allow federates to be more efficient if we allow access to handles.**

Next Steps

- **Place Java API on the DMSO home page for review**
- **The API is in HTML form for browsing**
- **Begin implementation of Java RTI**